

*Инструментальные  
средства  
информационных  
технологий*

*Проаналізовано відомі протоколи адаптивного HTTP мовлення та запропоновано алгоритм динамічної зміни якості відео в залежності від ширини Інтернет каналу.*

© О.В. Галкін, 2012

УДК 004.7

О.В. ГАЛКІН

**ДИНАМІЧНА ЗМІНА ЯКОСТІ  
ВІДЕО  
В ЗАЛЕЖНОСТІ ВІД ШИРИНИ  
ІНТЕРНЕТ-КАНАЛУ**

**Вступ.** Добре відомо, що основна проблема трансляції відео контенту через мережу Інтернет, це проблема «Останньої милі». Варіанти вирішення даної проблеми викладені в концепції адаптивного мовлення. На відміну від традиційного мовлення, в процесі відтворення контенту через мережу Інтернет, враховуються умови в яких знаходиться кінцевий користувач, тобто його поточна пропускна швидкість мережі. Якщо швидкість з'єднання недостатня, то якість зображення зменшується, і навпаки, якщо є можливість перегляду медіа кращої якості, то відбувається перехід на вищу якість даних. Існує багато варіантів реалізації технології адаптивного мовлення, однією з цікавих є трансляція на сервері потокового медіа двох потоків певного контенту з різними показниками якості та використання кадрів з потоку вищої якості та нижчої в залежності від умов на стороні клієнта [1]. Не менш цікавим є варіант трансляції контенту, використовуючи кодування в реальному часі, де зміна показників якості контролюється ключовим кадром, за допомогою якого можливо проводити кодування з будь-якої частини медіа контенту [2]. Але така можливість потребує надто багато ресурсів сервера. Існує також варіант використання специфічного контейнера для зберігання відеоданих в декількох потоках з можливостями компресії. Даний контейнер має назву MVC (Multiview Video Coding), який входить до набору алгоритмів H.264/AVC, що запатентовані і не доступні для широкого

використання [3]. Далі більш докладно проаналізуємо ві-домі на сьогодні технології адаптивного HTTP мовлення.

**Аналіз технологій адаптивного HTTP-мовлення.** На сьогоднішній день існує 4 основних технології доставки контенту на абонентські пристрої через мережу Інтернет, які використовуються різними операційними системами:

- технологія HLS;
- технологія WebM;
- технологія Silverlight Smooth Streaming;
- технологія HTTP dynamic Streaming.

Технологія HTTP Live Streaming (HLS) належить компанії Apple. Процес організації мовлення за протоколом HLS можна представити таким чином:

виконується стиснення відео сигналу, отриманого з прямої трансляції або з файлу, в формат H.264/TS з різним вихідним бітрейтом (використовуючи різні профілі стиснення);

отриманий потік сегментується для отримання коротких «шматків» контенту, довжиною, як правило, 10 секунд кожен;

генерується файл-плейлист (у форматі m3u або m3u8), який містить посилання на кожен сегмент потоку;

користувач виконує завантаження спочатку плейлиста, а потім сегментованого потоку з звичайного HTTP-сервера.

Технологія HLS дозволяє виконувати мовлення з бітрейтом, який адаптивно змінюється, при цьому рішення про те, потік якої якості слід завантажувати в даний момент часу, приймає кінцеве обладнання користувача, а не відео сервер. Рішення приймається на підставі доступної смуги пропускання, що дозволяє передавати через Інтернет відео з високим рівнем якості:

для кожного каналу або файлу генерується індексний файл із зазначенням різних профілів (відповідних потокам різної якості);

кінцевий користувальницький пристрій вибирає потік з найбільш підходящим бітрейтом на підставі того, скільки часу потрібно для завантаження одного сегмента;

кожен сегмент містить 10 секунд відео, тому приймальний пристрій може автоматично з інтервалом 10 секунд адаптуватися до змінних параметрів передачі.

Дана технологія має ряд переваг. HLS є простим і ефективним рішенням для організації адаптивного мовлення у відкритих мережах. Підтримку даного стандарту легко реалізувати на приймальних пристроях, що сприяє його широкому впровадженню в майбутньому. Багато виробників мікросхем нині можуть надати апаратні рішення для декодування H.264, що дозволяє використовувати стандарт в мобільних пристроях завдяки низькому енергоспоживанню і невеликому навантаженню на ЦП. Нарешті, використання контейнера MPEG-2 TS значно спрощує інтеграцію HLS в існуючі системи цифрового телебачення.

Smooth Streaming [4] – це протокол мовлення відео контенту, який представлений як розширення технології Silverlight 3.0. Дана технологія працює з

контейнером PIFF (Protected Interoperable File Format), при цьому загальні принципи роботи протоколу дуже схожі на функціонування HLS: відео та аудіо інформація стискається з використанням кодеків H.264 і AAC (або VC-1/WMA) відповідно, отримані потоки сегментуються, мультиплекуються в контейнер PIFF і поширюються за протоколом HTTP через Web-сервер і CDN. Для отримання відео потоку користувач спочатку завантажує з сервера файл-manifest у форматі XML, в якому міститься інформація про доступні аудіо і відео доріжки та їх бітрейти. Потім клієнт запитує один або більше сегментів, які передаються по верх протоколу HTTP. Подібно HLS, керування бітрейтом в Smooth Streaming здійснює клієнтська сторона. Однак, на відміну від HLS, де посилання на сегменти представлені в явному вигляді у файлі-плейлисті, файл-manifest протоколу MSS містить інформацію, яка дозволяє клієнтському пристрою згенерувати посилання на сегменти на підставі даних синхронізації, що містяться в потоці. Тому при мовленні live-потоків клієнту не потрібно періодично заново завантажувати файл-manifest. Перевагою даної технології є те, що Smooth Streaming підтримує наявність декількох аудіо доріжок і декількох доріжок субтитрів у потоці. Для отримання найбільш якісного зображення, плеєр Microsoft Silverlight перед вибором потоку з великим бітрейтом аналізує можливості пристрою. Також Smooth Streaming підтримує інтеграцію DRM, крім Microsoft PlayReady. Позитивним також є той факт, що компанія Microsoft надає дуже докладні специфікації з великою кількістю прикладів, що значно полегшує розуміння функціонування та впровадження технології.

Технологія HTTP Dynamic Streaming (HDS) представлена компанією Adobe. Технічно, принцип функціонування HDS дуже схожий на принцип роботи протоколу Microsoft Smooth Streaming. Відео потік складається з: manifest-файла у форматі XML з розширенням. F4m; сегментованих файлів, які містять фрагменти потоку MPEG-4 і мають розширення. f4f; індексних файлів з розширенням. F4x, які містять специфічну інформацію про фрагменти, розташовані всередині сегментованих файлів. HDS підтримує відео кодеки H.264 і VP6 і аудіо кодеки AAC і MP3.

Перевага цієї технології полягає в тому, що практично будь-який комп'ютер у світі може відтворювати потоки HTTP Dynamic Streaming.

Технологія WebM представлена компанією Google. Функціонування стандарту WebM відрізняється від інших технологій. При його використанні не потрібно сегментування вихідної інформації, тому з позиції WebM один медіапотік представляється у вигляді одного файлу. Мовлення live-потоків або VOD-файлів за протоколом WebM здійснюється наступним чином: відео та аудіо контент стискається за допомогою кодеків VP8 і Vorbis відповідно (одні й ті ж потоки або файли кодуються з різним бітрейтом); стиснений контент мультиплекується в файл, який має автоматично оновлюватися в випадку live-мовлення; отриманий WebM-файл потрапляє до абонента через HTTP-сервер.

Необхідно відзначити, що в даному випадку підвищується складність реалізації live-мовлення, тому мультиплексування потоків у контейнер має виконувати

тися безперервно і кінцевий файл постійно змінюється з часом. Це призводить до того, що завдання кешування потоків, переданих за протоколом WebM, набагато складніше, ніж кешування окремих сегментів відео файлів. Проте даний факт одночасно є перевагою WebM, тому що його можна безпосередньо використовувати як формат зберігання файлів.

Запропонований далі підхід, дозволяє реалізувати технологію адаптивного мовлення без використання спеціалізованих серверів відповідних компаній. В його основі лежить принцип використання технології передачі відео за протоколом RTMP та зміна характеристик відео потоку в залежності від умов користувача, а в подальшому перенесення проміжної обробки на клієнтську сторону за допомогою технології псевдо потоків. Окрім того, запропонована система також включає програмне забезпечення для автоматичної обробки відео контенту на основі статистичних даних, отриманих від користувачів даного ресурсу.

**Система автоматичної підготовки контенту.** Основна ідея системи полягає у тому щоб надати користувачеві якісно підготовлений контент для перегляду в режимі он-лайн, зважаючи на пропускну можливість його мережі. Одним із можливих варіантів реалізації даної ідеї є підготовка відеоматеріалів з різними показниками якості, щоб можливо було передавати контент по мережі без затримок. Зважаючи на проблему «останньої милі» також необхідно контролювати процес відтворення для оперативного реагування на зміну пропускну швидкості каналу мережі.

Система складається з двох частин. Перша частина відповідає за доставку контенту кінцевому користувачеві, контроль за зміною пропускну швидкості мережі та відповідне реагування на ці зміни. Друга частина являє собою програмний комплекс, який створений для полегшення процесу додавання нового контенту на сервер та його попередню обробку. Також він відповідає за обробку статистичних даних та автоматичну підготовку відеоматеріалів у залежності від потреб користувачів.

Серверна частина написана у вигляді Java Enterprise проекту, з'єднання з базою даних забезпечується за допомогою технології EJB. Під час входу на початкову сторінку проекту відбувається вимірювання пропускну швидкості мережі клієнта, на основі отриманих даних генеруються посилання на контент, який в даний момент знаходиться на сервері. Вибір оптимального варіанта відео файла здійснюється на стороні сервера, servlet на основі отриманих даних проводить підбір розміру файла, який у даний момент наявний в системі. Якщо не знайдено жодного відео файла, який задовольнив би потреби користувача, то обирається найближчий, із наявних даних, до отриманого запиту зі сторони користувача. Дані про пропускну спроможність та дату відвідання сайту заносяться до бази даних для подальшої обробки та підготовки відповідних матеріалів. Під час перегляду наданого контенту може виникнути ситуація коли швидкість мережі може змінитись і користувачеві буде некомфортно переглядати обраний матеріал, саме на цю ситуацію орієнтований набір коду який виконується на стороні користувача і відслідковує такого роду зміни. В даній реалізації присутня система перевірки в основі якої лежить відслідковування часу прогнозованого відтво-

рення в порівнянні з фактичним часом, впродовж якого відбувалося відтворення. Якщо фактичні й прогнозовані результати не збігаються, користувачеві автоматично проводиться зміна контенту на один із тих, що на даний момент наявні в системі, якщо такий матеріал відсутній, то зміна не проводиться, а в базу даних додаються дані про необхідність підготовки відео з відповідними показниками якості. Зміна відеороликів «на льоту» є однією з основних проблем відтворення контенту, тому що необхідно приховати сам факт зміни від користувача. Дану проблему можна вирішити лише за рахунок медіа сервера, додавши до нього можливість попереднього завантаження фрагментів даних до початку їх відтворення. У разі використання комерційного варіанту WOWZA Media Server, то тут лише присутня можливість створити додаток до сервера, використовуючи відкрите API для розробників. У випадку Open Source проектів, таких як Red5 Media Server, залишається лише можливість редагування вихідних кодів та ручного збирання проекту.

Під час розробки другої частини проекту було реалізовано програмний інтерфейс до командного рядка FFmpeg, його основна задача полягає у підготовці командного рядка в залежності від заданих параметрів якості відповідного контенту. В процесі розробки виникла необхідність зчитування метаданих (довжина ролику, бітрейт, розмір файлу, роздільна здатність), для вирішення цієї проблеми використано також FFmpeg, але в режимі зчитування інформації про файл. Щоб отримати ці дані використовується перенаправлення потоків стандартного вводу/виводу (stdin/stdout) та подальший розбір отриманих даних. Інформація про новий підготовлений контент заноситься до бази даних і стає доступною для використання. У даній частині проекту реалізований прямий доступ до бази даних, що спрощує передачу інформації між сервером бази даних та даної програми. Програма виконується постійно з заданим інтервалом перевіряє базу даних на наявність нових запитів, якщо такі присутні, то вона починає підготовку і кодування відповідних відео файлів. Паралельно з потоком в якому здійснюються аналіз та обробка, працює потік який обробляє команди користувача. Завдяки цим командам користувачеві надана можливість керування обробкою інформації. Для задоволення потреб користувачів система дозволяє автоматично готувати необхідний матеріал на основі аналізу отриманої статистики. В процесі активності користувача збирається інформація про переглянуті ним відеоролики, пропускну швидкість його Інтернет з'єднання та кількість запитів на зміну якості матеріалу. Отримані дані сортуються за спаданням від найпопулярнішого контенту до менш популярного, таким чином готується в першу чергу матеріал найбільш затребуваний користувачами. Процес підготовки починається відразу після перевірки наявності необхідної інформації в системі, інтервал перевірки задається адміністратором системи в залежності від навантаження.

Отримання метаданих з файлу здійснюється шляхом запуску утиліти FFmpeg у режимі зчитування. Щоб провести всі необхідні маніпуляції відбувається захоплення вихідного потоку утиліти та його подальший розбір з перетворенням отриманої інформації у необхідну форму. Перекодування проходить за допомогою утиліти FFmpeg, на вхід якої подається підготовлений командний

рядок, що генерується в автоматичному режимі та включає всю необхідну інформацію як про вхідні дані, так і про вихідні.

**Висновки.** Проаналізовано відомі 4 системи адаптивного HTTP мовлення та продемонстровано один із варіантів реалізації концепції адаптивного мовлення за допомогою сервера потокового медіа, використовуючи протокол RTMP. Ключовою відмінністю даного підходу є те, що для організації адаптивного мовлення не потрібно використовувати спеціалізовані Веб-сервери, використання та обслуговування яких потребує значних капіталовкладень. Запропоновано варіант системи автоматичного адміністрування та підготовки медіа контенту на основі аналізу статистичних даних зібраних іншою частиною проекту, яка спрощує адміністрування та дозволяє раціонально використовувати простір носіїв зберігання інформації.

1. Xiaoyan Sun, Feng Wu, Shipeng Li, Wen Gao, Ya-Qin Zhang «Seamless switching of scalable video bitstreams for efficient streaming» // IEEE Transaction on multimedia. – 2004. – Vol. 6, N 2. – P. 291 – 303.
2. Xiaoyan Sun, Feng Wu, Shipeng Li, Guobin Shen, Wen Gao «Drift-Free Switching of Compressed Video Bitstreams at Predictive Frames» // IEEE Trans. on Circuits Systems for Video Technology. – 2006. – Vol. 16, N 5. – P. 565 – 576.
3. Xun Guo, Yan Lu, Feng Wu, Wen Gao, Shipeng Li «Free Viewpoint Switching in Multi-view Video Streaming Using Wyner-Ziv Video Coding» // SPIE Visual Communications and Image Processing, VCIP 2006, San Jose, CA, USA, Jan. 2006
4. Microsoft IIS Smooth Streaming technology // [http://jmvvm.vse.cz/wp-content/uploads/2011/05/t\\_IIS\\_Smooth\\_Streaming\\_Technical\\_Overview.pdf](http://jmvvm.vse.cz/wp-content/uploads/2011/05/t_IIS_Smooth_Streaming_Technical_Overview.pdf)

Одержано 15.10.2012

*A.V. Галкин*

#### ДИНАМИЧЕСКОЕ ИЗМЕНЕНИЕ КАЧЕСТВА ВИДЕО В ЗАВИСИМОСТИ ОТ ШИРИНЫ ИНТЕРНЕТ-КАНАЛА

Проанализировано известные протоколы HTTP вещания и предложено реализацию алгоритма изменения качества видео в зависимости от ширины Интернет канала.

*O.V. Galkin*

#### DYNAMIC CHANGE OF QUALITY OF VIDEO DEPENDING ON WIDTH OF THE INTERNET-CHANNEL

We analyzed famous protocols HTTP broadcast and proposed the algorithm of changing quality of video depend on width of internet channel.

#### **Про автора:**

*Галкін Олександр Володимирович,*  
доцент кафедри інформаційних систем факультету кібернетики

О.В. ГАЛКІН

---

Київського національного університету імені Тараса Шевченка.